

Fast multilevel sparse Gaussian kernels for high-dimensional approximation and integration

Zhaonan Dong, Emmanuil H. Georgoulis, Jeremy Levesley, and Fuat Usta

Department of Mathematics, University of Leicester,
University Road, Leicester, LE1 7RH, United Kingdom
{ zd14, Emmanuil.Georgoulis, j.levesley, fu6 } @le.ac.uk

Abstract. A fast multilevel algorithm based on directionally scaled tensor-product Gaussian kernels on structured sparse grids is proposed for interpolation of high-dimensional functions and for the numerical integration of high-dimensional integrals. The algorithm is based on the recent Multilevel Sparse Kernel-based Interpolation (MLSKI) method (Georgoulis, Levesley & Subhan, *SIAM J. Sci. Comput.*, 35(2), pp. A815–A831, 2013), with particular focus on the fast implementation of Gaussian-based MLSKI for interpolation and integration problems of high-dimensional functions $f : [0, 1]^d \rightarrow \mathbb{R}$, with $5 \leq d \leq 10$. The MLSKI interpolation procedure is shown to be interpolatory and a fast implementation is proposed. More specifically, exploiting the tensor-product nature of anisotropic Gaussian kernels, one-dimensional cardinal basis functions on a sequence of hierarchical equidistant nodes are precomputed to machine precision, rendering the interpolation problem into a fully parallelisable ensemble of linear combinations of function evaluations. A numerical integration algorithm is also proposed, based on interpolating the (high-dimensional) integrand. A series of numerical experiments highlights the applicability of the proposed algorithm for interpolation and integration for up to 10-dimensional problems.

1 Introduction

Approximation, interpolation and numerical integration of high-dimensional functions have numerous applications in applied sciences, ranging from mathematics, statistics, physics and engineering, to economics and finance.

For low dimensional multivariate functions $f : \Omega \rightarrow \mathbb{R}$, $\Omega \subset \mathbb{R}^d$, with $2 \leq d \leq 4$, classical interpolation and integration (cubature) methods are able to deliver accurate and efficient results to high accuracy, see e.g., [7, 25] and the references therein. For $d \geq 5$, standard algorithms are challenged by the, so-called, curse of dimensionality. Indeed, standard polynomial, spline or kernel-based interpolation methods of a function $f : [0, 1]^d \rightarrow \mathbb{R}$, on a d -dimensional grid with N nodes in each direction, typically require N^d function evaluations to achieve rates of convergence of order $N^{-\alpha}$, for some $\alpha > 0$ independent of d ; a manifestation of

the curse of dimensionality, as the interpolation problem becomes exponentially more computationally intensive with d .

To address this issue, a number of methods based on either randomness, especially for integration (Monte-Carlo, quasi-Monte-Carlo, multilevel Monte-Carlo; see, e.g., [5,9,19,6,20] and the references therein) or on Smolyak/sparse grid/hyperbolic cross/boolean interpolation-type constructions (see, e.g., [24,8,26,23,4] and the references therein) have appeared in the literature during the last five decades. Monte-Carlo-type approaches are, generally speaking, robust and easy to implement and their accuracy is independent of the problem dimension d , at the cost of typically slow convergence in a probabilistic sense [5,9]. Sparse grids and hyperbolic crosses are applicable when the function f has sufficiently regular mixed derivatives; in such cases, they can deliver, typically algebraic, rates of convergence, independent of d with typical complexity of $N(\log N)^{d-1}$, where N is the number of points in each direction. More recently, “ p -version” sparse grids, based on polynomial interpolation on Chebyshev or Clenshaw-Curtis sparse tensor-product type grids have been proposed for the calculation of high-dimensional integrals arising in the numerical approximation of elliptic boundary-value problems with random coefficients [2,22,3].

Another approach, that can be positioned within the second class of methods described above is the recent Multilevel Sparse Kernel-based Interpolation (MLSKI) method [14]. MLSKI method is based on the concept of combination (also known as d -dimensional boolean interpolation [8]) of smaller interpolation problems on directionally uniform nodes, whose union coincides with the sparse grid node systems produced by hierarchical linear splines. Each interpolation sub-problem is evaluated using anisotropically scaled versions of translation of standard kernels/radial basis functions. All interpolation problems are then linearly combined appropriately, leading to the Sparse Kernel-based Interpolation method (SKI). To further exploit the approximation power of the SKI interpolants, which use naturally nested sparse grid nodes, the interpolation procedure is performed in a multilevel fashion, starting from a coarse SKI interpolant s_0 of the function f , and then continuing by interpolating the residual $f - s_0$ on the next level sparse grid, and so on. We stress that, despite the fact that the MLSKI method is applied on sparse grids arising from *uniform* full grids, it has not been observed to be susceptible to classical Runge-type instabilities. Hence, MLSKI can be viewed as an alternative to “ p -version” sparse grids, by offering comparable convergence rates on “uniform” (and, therefore, hierarchical) sparse grids.

This work is concerned with further investigating the MLSKI approach both theoretically and numerically. More specifically, focusing on MLSKI using Gaussian kernels, we show that at each level of SKI, the method indeed, interpolates the function f . To achieve numerical exactness on all the nodes, the MLSKI approach it is sufficient to use tensor-product-type kernels, i.e., Gaussians. This also leads naturally to the fast implementation of the sub-problem computations of MLSKI [14], each of which can be computed completely independently. In particular, exploiting the tensor-product nature of anisotropic Gaussian kernels,

one-dimensional cardinal basis functions on a sequence of hierarchical equidistant nodes are precomputed to machine precision, rendering the interpolation problem into a fully parallelisable ensemble of linear combinations of function evaluations. A numerical integration algorithm is also proposed, based on interpolating the (high-dimensional) integrand. A series of numerical experiments is presented, highlighting the practical applicability of the proposed algorithm for interpolation and integration for up to 10-dimensional problems.

In Section 2, we give a brief description of the method, while, in Section 3, we show that for the Gaussian kernel the cardinal functions for interpolation can be written as a product of univariate cardinal functions. In Section 4, we show that if the kernel is a tensor product (such as the Gaussian) then MLSKI is also an interpolatory scheme. In Section 5, we discuss the integration scheme with some numerical results which demonstrate the good approximation properties of the method.

2 Multilevel sparse kernel-based interpolation

We begin by briefly reviewing the MLSKI method, introduced in [14]. To construct the sparse kernel-based interpolant, at each level, we solve a number of anisotropic radial basis function interpolation problems on appropriate direction-wise uniform sub-grids. We briefly note that due to the careful selection of anisotropic scalings for the individual interpolation problems, these are typically sufficiently well conditioned for practical computations; cf. [14]. The individually solved interpolation problems are then linearly combined to obtain the sparse kernel-based interpolant. The multilevel sparse kernel based interpolation (MLSKI) uses a residual interpolation at different levels.

More specifically, let $\Omega := [0, 1]^d$, $d \geq 2$, and let $u : \Omega \rightarrow \mathbb{R}$. For a multi-index $\mathbf{l} = (l_1, \dots, l_d) \in \mathbb{N}^d$, we define the family of directionally uniform grids $\{\mathbb{X}_{\mathbf{l}} : \mathbf{l} \in \mathbb{N}^d\}$, in Ω , with meshsize $h_{\mathbf{l}} = 2^{-\mathbf{l}} := (2^{-l_1}, \dots, 2^{-l_d})$. Then, $\mathbb{X}_{\mathbf{l}}$ consists of the points $\mathbf{x}_{\mathbf{l}, \mathbf{i}} := (x_{l_1, i_1}, \dots, x_{l_d, i_d})$, with $x_{l_j, i_j} = i_j 2^{-l_j}$, for $i_j = 0, 1, \dots, 2^{l_j}$, $j = 1, \dots, d$. The number of nodes $N^{\mathbf{l}}$ in $\mathbb{X}_{\mathbf{l}}$ is given by

$$N^{\mathbf{l}} = \prod_{j=1}^d (2^{l_j} + 1).$$

If $h_{l_j} = 2^{-n}$, for all $j = 1, \dots, d$, $\mathbb{X}_{\mathbf{l}}$ is the uniform *full grid* of level n , having size $N = (2^n + 1)^d$; this will be denoted by $\mathbb{X}^{n,d}$.

We also consider the following subset of $\mathbb{X}^{n-d+1,d}$,

$$\tilde{\mathbb{X}}^{n,d} := \bigcup_{|\mathbf{l}|_1 = n + (d-1)} \mathbb{X}_{\mathbf{l}}, \quad (1)$$

with $|\mathbf{l}|_1 := l_1 + \dots + l_d$, which will be referred to as the *sparse grid of level n in d dimensions*; see Figure 1 an illustration with $n = 4$ and $d = 2$.

Further, we define the transformation matrix $A_{\mathbf{l}} \in \mathbb{R}^{d \times d}$ by

$$A_{\mathbf{l}} := \text{diag}(2^{l_1}, \dots, 2^{l_d}).$$



Fig. 1: Sparse grid $\tilde{\mathbb{X}}^{4,2}$ via (1).

The anisotropic *radial basis function* (RBF) interpolant S_{A_1} of u at the points of \mathbb{X}_1 is then defined by

$$S_{A_1}(\mathbf{x}) := \sum_{\mathbf{x}_{1,i} \in \mathbb{X}_1} c_{1,i} \varphi(\|A_1(\mathbf{x} - \mathbf{x}_{1,i})\|), \quad (2)$$

for $\mathbf{x} \in \Omega$, where $c_{1,i} \in \mathbb{R}$ are chosen so that the interpolation conditions

$$S_{A_1}|_{\mathbb{X}_1} = u|_{\mathbb{X}_1},$$

are satisfied. Although ϕ can be chosen from a large family of RBF kernels (cf. [14] for details), this work will be focused on the choice of Gaussian kernels, that is, $\phi(r) = \exp(-c^2 r^2)$, for $r \geq 0$ and $c > 0$, denoting the, so-called, *shape-parameter*.

The *sparse kernel-based interpolant* (SKI, for short) $\tilde{S}_{n,d}$ is then given by

$$\tilde{S}_{n,d}(\mathbf{x}) = \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \sum_{|\mathbf{l}|_1 = n+(d-1)-q} S_{A_1}(\mathbf{x}). \quad (3)$$

The above combination formula has been used, for instance, for Lagrange polynomial interpolation in [8], and for the numerical solution of elliptic partial differential equations using the finite element method on sparse grids in [15,13].

In Figure 2, we show a visualisation of the SKI interpolant $\tilde{S}_{4,2}$ as a linear combination of the constituent sub-grid interpolants. In the next section, we shall show that this procedure indeed results to an interpolation method, when ϕ is Gaussian.

The convergence of SKI was investigated in [14], where it was found that, although it results to acceptable approximation power when interpolating simpler tensor-product-type functions f , it can be also prone to very slow convergence on more challenging functions. This appears to be due to the fact that, in contrast with standard sparse grid/hyperbolic cross/Smolyak-type polynomial or linear spline interpolants, the approximation space of the SKI of level n does *not* contain the respective approximation spaces of levels $1, \dots, n-1$ as subspaces. In short, the SKI interpolation is not hierarchical in each level.

To overcome this, a multilevel approach to SKI was proposed in [14]. There is a growing volume of literature on multilevel methods for RBFs, e.g. [12,17,27,11,10,21,16,18]. The idea is to interpolate a high frequency residual with a rougher interpolant.

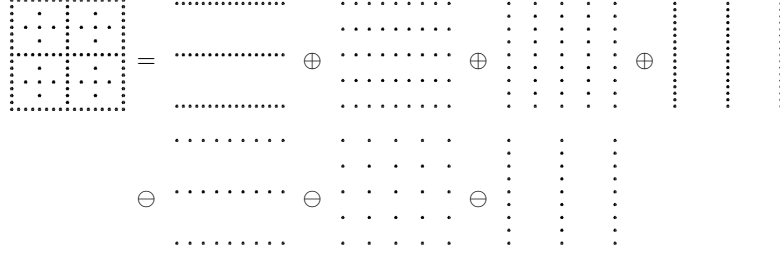


Fig. 2: The construction of $\tilde{S}_{4,2}$ interpolant on $\tilde{\mathbb{X}}^{4,2}$.

The setting of SKI is naturally suited to be used within a multilevel interpolation algorithm. Firstly, the sparse grids from lower to higher level are nested, i.e., $\tilde{\mathbb{X}}^{n,d} \subset \tilde{\mathbb{X}}^{n+1,d}$ for $n \in \mathbb{N}$. Secondly, each sub-grid interpolant uses appropriately scaled anisotropic basis function with the scaling being proportional to density of the corresponding constituent sub-grid. Finally, due to the geometrical progression in the problem size from one sparse grid to the next, a multilevel algorithm would not affect adversely the attractive complexity properties of SKI.

The *multilevel SKI* (MLSKI, for short) algorithm is initialised by computing the SKI $\tilde{S}_{n_0,d}$ on the coarsest sparse grid $\tilde{\mathbb{X}}^{n_0,d}$ and set $\Delta_0 := \tilde{S}_{n_0,d}$. Then, for $k = 1, \dots, n$, Δ_k is the sparse grid interpolant to the residual $u - \sum_{j=0}^{k-1} \Delta_j$ on $\tilde{\mathbb{X}}^{k,d}$. The resulting multilevel sparse kernel based interpolant is then given by

$$\tilde{S}_{n,d}^{\text{ML}} := \sum_{j=0}^n \Delta_j.$$

Below we describe how, if the kernel is a tensor product, we can construct a tensor product basis for the interpolation once and for all, and so avoid the costly solution of subsystems of equations, which is, in principle, highly scalable in a parallel computing architecture.

3 Tensor Products of Univariate Cardinal Functions

Gaussian kernels on \mathbb{R}^d can be viewed as tensor products of univariate Gaussians; this is evident also for the anisotropic versions of Gaussians, such as the ones discussed in the previous section. This crucial property of Gaussians will be instrumental in both the proof of interpolation of the MLSKI algorithm, and the development of fast procedures to evaluate the respective MLSKI interpolants.

We begin by considering the set of cardinal (also known as Lagrange) functions for each interpolation sub-problem in the SKI interpolant $S_{A_1}(\cdot)$ in (3), with the particular choice of Gaussian RBFs on the set \mathbb{X}_1 . Let $\chi_{\mathbf{l}, \mathbf{x}_{\mathbf{l}, \mathbf{i}}}$ be the cardinal function on the grid indexed by \mathbf{l} for the point $\mathbf{x}_{\mathbf{l}, \mathbf{i}} \in \mathbb{X}_1$. Let also $\chi_{l_j, x_{l_j, i_j}}$ denote the univariate cardinal function in one variable for the node x_{l_j, i_j} , with

respect to the set of nodes $\{x_{l_j, i_j} : i_j = 0, 1, \dots, 2^{l_j}\}$. Hence, for $j = 1, \dots, d$, there exist $\gamma_{l_j, i_j} \in \mathbb{R}$, such that

$$\chi_{l_j, x_{l_j, i_j}}(y) = \sum_{i_j=0}^{2^{l_j}} \gamma_{l_j, i_j} \exp(-c^2 h_{l_j}^2 (y - x_{l_j, i_j})^2),$$

for $y \in [0, 1]$. Then, we have

$$z(\mathbf{y}) := \prod_{j=1}^d \chi_{l_j, x_{l_j, i_j}}(y_j) = \sum_{\mathbf{x}_{\mathbf{l}, \mathbf{i}} \in \mathbb{X}_{\mathbf{l}}} \gamma_{\mathbf{l}, \mathbf{i}} \exp(-c^2 \sum_{j=1}^d h_{l_j}^2 (y_j - x_{l_j, i_j})^2),$$

where $\gamma_{\mathbf{l}, \mathbf{i}} = \prod_{j=1}^d \gamma_{l_j, i_j}$. This is exactly the form of the cardinal function based on the points in the grid $\mathbb{X}_{\mathbf{l}}$, which implies $z(\mathbf{y}) = \chi_{\mathbf{l}, \mathbf{x}_{\mathbf{l}, \mathbf{i}}}$, due to the uniqueness of Gaussian interpolation.

Hence, it is possible to compute the cardinal functions for multivariate approximation by computing *ab initio* (to arbitrarily high precision, e.g., by using symbolic calculators) the cardinal functions for univariate approximation up to (for instance) 5, 9, 17, \dots , 129 equally spaced points, and store these. The approximation process would then require no solution of linear systems, massively increasing the speed of the algorithm. This algorithm will be implemented below in the numerical experiments.

4 Tensor product kernels give interpolatory schemes

We shall show that the combination formula (3) is, indeed, an interpolant. To highlight the key ideas of the proof, we first consider an example in three dimensions. A two dimensional example is too straightforward, and a four dimensional one is already too complicated.

Setting, for instance, $d = 3$ and $n = 7$, we seek to compute the value of the sparse kernel-based interpolant $\tilde{S}_{3,7}$ to the function f at the point $a = (1/4, 1/8, 1/16) = (2^{-2}, 2^{-3}, 2^{-4})$, which first appears in the grid with multi-index $\mathbf{l} = (2, 3, 4)$. From (3), we see that for $d = 3$, the sub-grids of the previous two levels are linearly combined to give $\tilde{S}_{3,7}$. Hence, the set of multi-indices which index the approximation are $\{\alpha : |\alpha| = 7, 8, 9\}$.

We decompose these sets of multi-indices into two groups: The first sets of grids, let us call them Group 1, have two of the three components of the multi-index less than the corresponding components for the multi-index for the point, e.g., the grids represented by the multi-indices $\{\alpha = (k, 1, 1) : k = 5, 6, 7\}$. The remaining grids, let us call them Group 2, have only one component less than the corresponding component for the point; for instance the family of grids $\{\alpha = (k, l, 2) : k + l \leq 7, k \geq 2, l \geq 3\}$.

We shall now study the values of the function f on a typical set of grids from Group 1, $\{(k, 1, 1) : k = 5, 6, 7\}$. To this end, we consider the cardinal functions from points on these grids, and their values at the point a . Let $b =$

$(r_1 2^{-k}, r_2/2, r_3/2)$, for some $r_i \in \mathbb{N}_0$, $i = 1, 2, 3$, where $0 \leq r_1 \leq 2^k$, $0 \leq r_2, r_3 \leq 2$, with cardinal function $\chi_{(k,1,1),b}$. Then

$$\chi_{(k,1,1),b}(a) = \chi_{k,r_1 2^{-k}}(1/4) \chi_{1,r_2/2}(1/8) \chi_{1,r_3/2}(1/16) = 0,$$

if $r_1 2^{-k} \neq 1/4$, since $k \geq 2$. If $r_1 2^{-k} = 1/4$ then

$$\chi_{(k,1,1),b}(a) = \chi_{1,r_2/2}(1/8) \chi_{1,r_3/2}(1/16)$$

which is independent of k .

Thus $(1/4, r_2/2, r_3/2)$ is a node on all grids $\{\mathbb{X}_{(k,1,1)} : k = 5, 6, 7\}$, and the value of the cardinal functions $\chi_{(k,1,1),b}$ are identical at a . Hence, the contribution to the interpolant from $f(b)$, on these grids, is

$$\begin{aligned} & \sum_{q=0}^2 (-1)^q \binom{2}{q} f(b) \chi_{(7-q,1,1),b}(a) \\ &= f(b) \chi_{1,r_2/2}(1/8) \chi_{1,r_3/2}(1/16) \sum_{q=0}^2 (-1)^q \binom{2}{q} \\ &= 0. \end{aligned}$$

If we combine the results of the last two paragraphs we see that the contribution to the interpolant from all of the points on grids from Group 1 is 0.

We now turn to grids from Group 2. Let us consider points on the grids $\mathbb{X}_{(k,l,2)}$ for $k+l \leq 7$, $k \geq 2$, and $l \geq 3$, and their values at a . Let $b = (r_1 2^{-k}, r_2 2^{-l}, r_3/4)$, for some $0 \leq r_1 \leq 2^k$, $0 \leq r_2 \leq 2^l$, and $0 \leq r_3 \leq 4$ with cardinal function $\chi_{(k,l,2),b}$. Then,

$$\chi_{(k,l,2),b}(a) = \chi_{k,r_1 2^{-k}}(1/4) \chi_{l,r_2 2^{-l}}(1/8) \chi_{2,r_3/4}(1/16) = 0,$$

unless $r_1 2^{-k} = 1/4$ and $r_2 2^{-l} = 1/8$, since $k \geq 2$ and $l \geq 3$. If $r_1 2^{-k} = 1/4$ and $r_2 2^{-l} = 1/8$, then $\chi_{(k,l,2),b}(a) = \chi_{2,r_3/4}(1/16)$, which is independent of k and of l .

Thus, $(1/4, 1/8, r_3/4)$ is contained on all grids $\mathbb{X}_{(k,l,2)}$ for $k+l \leq 7$, $k \geq 2$, and $l \geq 3$, and the value of the cardinal functions $\chi_{(k,l,2),b}$, for all the permissible values of k are identical at a . Hence, the contribution to the interpolant from $f(b)$, on these grids, is

$$\begin{aligned} & \sum_{q=0}^2 (-1)^q \binom{2}{q} f(b) \sum_{k+l \leq 7-q, k \geq 2, l \geq 3} \chi_{(k,l,2),b}(a) \\ &= f(b) \chi_{2,r_3/4}(1/16) \sum_{q=0}^2 (-1)^q \binom{2}{q} \text{card} \{(k,l) : k+l \leq 7-q, k \geq 2, l \geq 3\} \\ &= f(b) \chi_{2,r_3/4}(1/16) \sum_{q=0}^2 (-1)^q \binom{2}{q} (3-q) = 0. \end{aligned}$$

Therefore, the contribution to the interpolant from all grids in Group 2 is also zero.

Thus, the only grid contributing to the interpolant is $\mathbb{X}_{(2,3,4)}$, and the only cardinal function from that grid which takes non zero values at a is the cardinal function at a itself. Therefore, the SKI $\tilde{S}_{7,3}$ is, indeed, an interpolant at all points of $\tilde{\mathbb{X}}^{7,3}$. Hopefully, this example highlights clearly the key role that tensor-product nature of the kernel has in the proof of interpolation property.

Equipped with the insight gained by the above example, we shall now consider the general case. To this end, we begin with the following counting result.

Lemma 1. *Let $\mathbf{k} \in \mathbb{N}^d$ and $|\mathbf{k}| < p \in \mathbb{N}$. Then,*

$$\text{card} \{ \mathbf{j} \in \mathbb{N}^d : \mathbf{j} \geq \mathbf{k}, |\mathbf{j}| = p \} = \binom{p - |\mathbf{k}| + d - 1}{d - 1}.$$

Proof: For $d = 1$, the result is immediate, since for $k < p$,

$$\text{card} \{ j \in \mathbb{N} : j \geq k, j = p \} = 1.$$

For $d \geq 2$, let us write $\mathbf{k} = (k_1, \tilde{\mathbf{k}})$, and consider we set , and consider $\{\tilde{\mathbf{j}} \in \mathbb{N}^{d-1} : |\tilde{\mathbf{j}}| = p - j_1\}$, for $j_1 = k_1, k_1 + 1, \dots, p - |\tilde{\mathbf{k}}|$. Then, by induction,

$$\text{card} \{ \tilde{\mathbf{j}} \in \mathbb{N}^{d-1} : \tilde{\mathbf{j}} \geq \tilde{\mathbf{k}}, |\tilde{\mathbf{j}}| = p - j_1 \} = \binom{p - j_1 - |\tilde{\mathbf{k}}| + d - 1}{d - 1}.$$

Thus,

$$\begin{aligned} \text{card} \{ \mathbf{j} \in \mathbb{N}^d : \mathbf{j} \geq \mathbf{k}, |\mathbf{j}| = p \} &= \sum_{j_1=k_1}^{p-|\tilde{\mathbf{k}}|} \text{card} \{ \tilde{\mathbf{j}} : \tilde{\mathbf{j}} \geq \tilde{\mathbf{k}}, |\tilde{\mathbf{j}}| = p - j_1 \} \\ &= \sum_{j_1=k_1}^{p-|\tilde{\mathbf{k}}|} \binom{p - j_1 - |\tilde{\mathbf{k}}| + d - 1}{d - 1} \\ &= \sum_{j_1=1}^{p-|\mathbf{k}|+1} \binom{j_1 + d - 2}{d - 1} \\ &= \binom{p - |\mathbf{k}| + d}{d}, \end{aligned}$$

using the well-known summation formula for binomial coefficients; e.g., [1, 1.2.2].

□

We are now ready to state and prove the main result of this section.

Theorem 1. *Assuming that the interpolation kernel has the form*

$$\psi(\mathbf{y}) = \prod_{i=1}^d \phi(y_i),$$

then SKI with this kernel is an interpolatory scheme.

Proof: We wish to compute the value of the sparse grid interpolant $\tilde{S}_{d,n}$ to the function f at the point $\mathbf{a} = (a_1 2^{-l_1}, a_2 2^{-l_2}, \dots, a_d 2^{-l_d})$, with $0 \leq a_j \leq 2^{l_j}$, $1 \leq j \leq d$, with at least one of the $a_j \in \mathbb{N}_0$ odd (this ensures that \mathbb{X}_1 is the grid with the smallest index in which this points appears). From hypothesis, the kernel is of tensor-product form, which implies the cardinal functions for interpolation are of the form

$$\chi_{\mathbf{m}, \mathbf{x}} = \prod_{j=1}^d \chi_{m_j, x_j} 2^{-m_j},$$

as observed in Section 3.

In order to create a decomposition of the indices as in the example above, we need to introduce some notation. Let $d_1, d_2 \in \mathbb{N}$, with $d_1 + d_2 = d$. Let $\mathbf{n}_1 \in \mathbb{N}^{d_1}$ with $\mathbf{n}_1(j) \in \{1, 2, \dots, d\}$, $j = 1, 2, \dots, d_1$, and $\mathbf{n}_1(j) < \mathbf{n}_1(j+1)$, $j = 1, 2, \dots, d_1 - 1$. Similarly, let $\mathbf{n}_2 \in \mathbb{N}^{d_2}$ with $\mathbf{n}_2(j) \in \{1, 2, \dots, d\}$, $j = 1, 2, \dots, d_2$, and $\mathbf{n}_2(j) < \mathbf{n}_2(j+1)$, $j = 1, 2, \dots, d_2 - 1$. Additionally, $\mathbf{n}_1(j) \neq \mathbf{n}_2(k)$ for any j, k . In other words, the components of \mathbf{n}_1 and \mathbf{n}_2 exhaust the set $\{1, 2, \dots, d\}$, and these numbers are ordered within the vectors \mathbf{n}_1 and \mathbf{n}_2 . We should note that once \mathbf{n}_1 is specified, \mathbf{n}_2 is uniquely determined and vice versa.

Let $\mathbf{m}_i \in \mathbb{N}^{d_i}$, $i = 1, 2$. Then, let the multi-index $\mathbf{m} = (\mathbf{n}_1, \mathbf{m}_1, \mathbf{n}_2, \mathbf{m}_2) \in \mathbb{N}^d$ have components $\mathbf{m}(k) = \mathbf{m}_i(j)$ if $\mathbf{n}_i(j) = k$, $i = 1, 2$. So, for instance, if $\mathbf{n}_1(3) = 7$, then $\mathbf{m}(7) = \mathbf{m}_1(3)$, and if $\mathbf{n}_2(4) = 5$, then $\mathbf{m}(5) = \mathbf{m}_2(4)$. In this way, we break multi-indices into two pieces in a convenient fashion.

On the other hand, for $\mathbf{m} \in \mathbb{N}^d$, let $\mathbf{m}_{\mathbf{n}_1} \in \mathbb{N}^{d_1}$ with $\mathbf{m}_{\mathbf{n}_1}(i) = \mathbf{m}(\mathbf{n}_1(i))$, $i = 1, 2, \dots, d_1$ and $\mathbf{m}_{\mathbf{n}_2} \in \mathbb{N}^{d_2}$ with $\mathbf{m}_{\mathbf{n}_2}(i) = \mathbf{m}(\mathbf{n}_2(i))$, $i = 1, 2, \dots, d_2$. Then, we have the identity

$$\mathbf{m} = (\mathbf{n}_1, \mathbf{m}_{\mathbf{n}_1}, \mathbf{n}_2, \mathbf{m}_{\mathbf{n}_2}).$$

Now, for each $d_1 = 1, 2, \dots, d-1$, and $\mathbf{n}_1 \in \mathbb{N}^{d_1}$, let

$$I(\mathbf{l}, d_1, \mathbf{n}_1) = \{\mathbf{m}_1 : \mathbf{m}_1 < \mathbf{l}_{\mathbf{n}_1}\}.$$

For each $\mathbf{m}_1 \in I(\mathbf{l}, d_1, \mathbf{n}_1)$, define

$$J(\mathbf{l}, d_1, \mathbf{n}_1, \mathbf{m}_1) = \{(\mathbf{n}_1, \mathbf{m}_1, \mathbf{n}_2, \mathbf{m}_2) : \mathbf{m}_2 \in \mathbb{N}^{d_2}, \mathbf{m}_2 \geq \mathbf{l}_{\mathbf{n}_2}\}.$$

The sets $J(\mathbf{l}, d_1, \mathbf{n}_1, \mathbf{m}_1)$ partition all multi-indices into sets with a fixed set of components of the multi-index less than those of \mathbf{l} and the remaining components greater than or equal to those of \mathbf{l} . The only multi-index missing from this set is \mathbf{l} itself.

We compute the cardinality of the subsets of $J(\mathbf{l}, d_1, \mathbf{n}_1, \mathbf{m}_1)$, given by

$$J(\mathbf{l}, d_1, \mathbf{n}_1, \mathbf{m}_1, q) = \{\mathbf{m} \in J(\mathbf{l}, d_1, \mathbf{n}_1, \mathbf{m}_1) : |\mathbf{m}| = n + d - 1 - q\},$$

for $q = 0, 1, \dots, d-1$, which, using Lemma 1, is given by

$$\text{card } J(\mathbf{l}, d_1, \mathbf{n}_1, \mathbf{m}_1, q) = \binom{n + d + d_1 - 2 - |\mathbf{l}_{\mathbf{n}_2}| - q}{d_1 - 1}. \quad (4)$$

We now consider the contribution to the interpolant from a typical point on one of the grids indexed by elements of $J(\mathbf{l}, d_1, \mathbf{n}_1, \mathbf{m}_1)$.

Let $\mathbf{m} = (\mathbf{n}_1, \mathbf{m}_1, \mathbf{n}_2, \mathbf{m}_2) \in J(\mathbf{l}, d_1, \mathbf{n}_1, \mathbf{m}_1)$. Then, by definition, $\mathbf{l}_{\mathbf{n}_2} \geq \mathbf{m}_2$. Let $\mathbf{x} = (x_1 2^{-m_1}, x_2 2^{-m_2}, \dots, x_d 2^{-m_d})$, for some $0 \leq x_j \leq 2^{m_j}$, $j = 1, 2, \dots, d$, with cardinal function $\chi_{\mathbf{m}, \mathbf{x}}$. Then,

$$\chi_{\mathbf{m}, \mathbf{x}}(\mathbf{a}) = \prod_{i=1}^d \chi_{m_i, x_i 2^{-m_i}}(a_i 2^{-l_i}).$$

Hence, $\chi_{\mathbf{m}, \mathbf{x}}(\mathbf{a}) \neq 0$ only if $x_i 2^{-m_i} = a_i 2^{-l_i}$, $i = \mathbf{n}_2(j)$, $j = 1, 2, \dots, d_2$. In this case,

$$\chi_{\mathbf{m}, \mathbf{x}}(\mathbf{a}) = \prod_{j=1}^{d_1} \chi_{m_{\mathbf{n}_1(j)}, x_{\mathbf{n}_1(j)} 2^{-m_{\mathbf{n}_1(j)}}}(a_{\mathbf{n}_1(j)} 2^{-l_{\mathbf{n}_1(j)}}),$$

which is independent of \mathbf{m}_2 .

So we have, for fixed \mathbf{n}_1 and \mathbf{m}_1 , the same contribution at \mathbf{a} from any individual point which appears in a grid in $J(\mathbf{l}, d_1, \mathbf{n}_1, \mathbf{m}_1)$. Thus, the contribution to from \mathbf{x} at point \mathbf{a} is

$$\begin{aligned} & \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} f(\mathbf{x}) \sum_{\mathbf{m} \in J(\mathbf{l}, d_1, \mathbf{n}_1, \mathbf{m}_1, q)} \chi_{\mathbf{m}, \mathbf{x}}(\mathbf{a}) \\ &= f(\mathbf{x}) \prod_{j=1}^{d_1} \chi_{m_{\mathbf{n}_1(j)}, x_{\mathbf{n}_1(j)} 2^{-m_{\mathbf{n}_1(j)}}}(a_{\mathbf{n}_1(j)} 2^{-l_{\mathbf{n}_1(j)}}) \\ & \quad \times \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \text{card } J(\mathbf{l}, d_1, \mathbf{n}_1, \mathbf{m}_1, q) \\ &= f(\mathbf{x}) \prod_{j=1}^{d_1} \chi_{m_{\mathbf{n}_1(j)}, x_{\mathbf{n}_1(j)} 2^{-m_{\mathbf{n}_1(j)}}}(a_{\mathbf{n}_1(j)} 2^{-l_{\mathbf{n}_1(j)}}) \\ & \quad \times \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \binom{n+d+d_1-2-|\mathbf{l}_{\mathbf{n}_2}|-q}{d_1-1} \\ &= 0, \end{aligned}$$

since the binomial coefficient is a polynomial in q of degree less than $d-1$, which is thus annihilated by the difference operator.

Thus, we see that all contributions from points in other grids other than X_1 are zero at points in X_1 . Clearly, the only contribution at $\mathbf{a} \in X_1$ from points in X_1 will be from \mathbf{a} itself. \square

5 Numerical Integration

Equipped with an interpolation method, we can now also discuss integration over the unit cube. A quadrature rule will approximate the integral

$$\int_{[0,1]^d} f(\mathbf{x}) d\mathbf{x} \approx \sum_{\mathbf{z} \in Z} w_{\mathbf{z}} f(\mathbf{z}),$$

where $Z \subset [0,1]^d$ is a finite set of points, and $\{w_{\mathbf{z}} : \mathbf{z} \in Z\}$ are the *weights*. If we have a cardinal basis for Z , $\{\chi_{\mathbf{z}} : \mathbf{z} \in Z\}$, then we approximate

$$f = \sum_{\mathbf{z} \in Z} f(\mathbf{z}) \chi_{\mathbf{z}},$$

giving rise to the quadrature formula

$$\int_{[0,1]^d} f(\mathbf{x}) d\mathbf{x} \approx \sum_{\mathbf{z} \in Z} f(\mathbf{z}) \int_{[0,1]^d} \chi_{\mathbf{z}}(\mathbf{x}) d\mathbf{x}.$$

Thus, the weights are given by

$$\int_{[0,1]^d} \chi_{\mathbf{z}}(\mathbf{x}) d\mathbf{x}.$$

In the case when the cardinal functions are tensor products, viz., $\chi_{\mathbf{z}} = \chi_{z_1}^1 \times \cdots \times \chi_{z_d}^d$ (as is the case of Gaussians, as observed in Section 3), we can straightforwardly compute the weights as follows:

$$\int_{[0,1]^d} \chi_{\mathbf{z}}(\mathbf{x}) d\mathbf{x} = \prod_{j=1}^d \int_0^1 \chi_{z_j}^j(x_j) dx_j.$$

As also mentioned in Section 3, we can compute the cardinal functions offline to arbitrary precision for, e.g., up to 129 equally spaced points, thereby allowing for fast and highly accurate computation of the quadrature weights. Indeed, the integral of a univariate cardinal function is of the form

$$I = \int_0^1 \chi(x) dx = \int_0^1 \sum_{i=1}^N c_i \exp(-r^2(x - y_i)^2) dx,$$

for some $r, y_i, c_i \in \mathbb{R}$, $i = 1, \dots, N$, which, upon the change of variables $u = r(x - y_i)$, gives

$$I = \frac{1}{r} \sum_{i=1}^N c_i \int_{-ry_i}^{1-ry_i} \exp(-u^2) du = \frac{\pi}{2r} \sum_{i=1}^N c_i (\operatorname{erf}(r(1 - y_i)) - \operatorname{erf}(-ry_i)),$$

with the error function erf defined by

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

The error function can be computed to arbitrary precision.

5.1 Examples

In this section we give a number of examples. We would like to develop a set of benchmark examples so we invite other researchers to contact us with their results for these cases, and perhaps forward us their own examples. We do two tensor product examples, respectively in 5 and 10 dimensions, and then two non-tensor product examples, the first being smooth, the second with derivative singularities on the boundary. The motivation for the second choice is approximation of the pay-off function in option pricing problems.

1. Consider the function

$$f(\mathbf{x}) = \prod_{i=1}^5 4x_i(1 - x_i). \quad (5)$$

The corresponding exact integral of this function on the domain $[0, 1]^5$ with 16 digits accuracy is $(\frac{2}{3})^5 = 0.131687242798354$. In Table 1 we give the MLSKI quadrature computation of the same integral, recording the number of nodes use, and the absolute and relative errors.

nodes	Absolute error	Relative error
243	3.0091e-2	2.2850e-1
1053	5.1232e-3	3.8904e-2
3753	1.3013e-3	9.8818e-3
12033	1.4927e-4	1.1335e-3
36033	3.6134e-5	2.7439e-4
102785	3.4530e-6	2.6222e-5
282525	8.1811e-7	6.2125e-6
754845	6.9041e-8	5.2428e-7

Table 1: MLSKI quadrature for f .

2. The following example is in 10 dimensions:

$$g(\mathbf{x}) = \prod_{i=1}^{10} e^{(-x_i(1-x_i))} \quad (6)$$

The corresponding exact integral of this function on the domain $[0, 1]^{10}$ with 16 digits accuracy is 0.194279067580947. Table 2 shows how the error reduces with respect to the increase of degrees of freedom.

nodes	Absolute-error	Relative-error
59049	1.5068e-1	7.7556e-1
452709	5.8153e-3	2.9933e-2
2421009	3.5882e-3	1.8469e-2
10819089	4.9348e-4	2.5400e-3

Table 2: MLSKI quadrature for g .

3. Let us now consider the non-tensor product Franke 4D function:

$$\begin{aligned}
u_{F4D}(\mathbf{x}) = & \frac{3}{4}e^{(-(9x_1-2)^2-(9x_2-2)^2-(9x_3-2)^2)/4-(9x_4-2)^2)/8} \\
& + \frac{3}{4}e^{(-(9x_1+1)^2)/49-((9x_2+1)^2)/10-((9x_3+1)^2)/29-((9x_4+1)^2)/39} \\
& + \frac{1}{2}e^{(-(9x_1-7)^2)/4-(9x_2-3)^2-((9x_3-5)^2)/2-((9x_4-5)^2)/4} \\
& - \frac{1}{5}e^{(-(9x_1-4)^2)/4-(9x_2-7)^2-((9x_3-5)^2)-((9x_4-5)^2)}.
\end{aligned} \tag{7}$$

The integral of this function on the domain $[0, 1]^4$ with 16 digits accuracy is 0.037221856819405. The errors in integration are shown in Table 3.

nodes	Absolute-error	Relative-error
81	1.6398e-2	4.4055e-1
297	1.2736e-2	3.4216e-1
945	7.9106e-3	2.1253e-1
2769	5.4904e-3	1.4751e-1
7681	5.5825e-4	1.4998e-2
20481	1.3012e-4	3.4959e-3
52993	1.6245e-5	4.3643e-4
133889	1.2027e-7	3.2312e-6
331777	2.2934e-8	6.1615e-7

Table 3: MLSKI quadrature for U_{F4D} .

4. The final example is a 5 dimensional non-tensor product function with derivative discontinuities in different directions on the boundary. In order to apply MLSKI to derivative pricing problems we would need to be able to approximate such functions as

$$F_{\text{payoff}}(\mathbf{x}) = \sum_{i=1}^5 \max(x_i - \frac{1}{2}, 0). \tag{8}$$

The exact integral of this function on the domain $[0, 1]^5$ is $\frac{5}{8}$. In Table 4 we see how the error behaves with regard to the degrees of freedom.

nodes	Absolute error	Relative error
243	1.5129e-1	2.4206e-1
1053	5.4282e-3	8.6851e-3
3753	2.9705e-3	4.7529e-3
12033	1.0128e-3	1.6206e-3
36033	3.2119e-4	5.1390e-4
102785	9.0693e-5	1.4511e-4
282525	2.2032e-5	3.5251e-5
754845	5.7779e-6	9.2447e-6

Table 4: MLSKI quadrature for F_{payoff} .

References

1. A. ABRAMOWITZ AND I. STEGUN, *Handbook of Mathematical Functions*, National Bureau of Standards, 1964.
2. I. BABUŠKA, F. NOBILE, AND R. TEMPONE, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM J. Numer. Anal., 45 (2007), pp. 1005–1034.
3. ———, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM Rev., 52 (2010), pp. 317–355.
4. H.-J. BUNGARTZ AND M. GRIEBEL, *Sparse grids*, Acta Numer., 13 (2004), pp. 147–269.
5. R. E. CAFLISCH, *Monte Carlo and quasi-Monte Carlo methods*, in Acta numerica, 1998, vol. 7 of Acta Numer., Cambridge Univ. Press, Cambridge, 1998, pp. 1–49.
6. K. A. CLIFFE, M. B. GILES, R. SCHEICHL, AND A. L. TECKENTRUP, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Comput. Vis. Sci., 14 (2011), pp. 3–15.
7. P. J. DAVIS, *Interpolation and approximation*, Dover Publications Inc., New York, 1975. Republication, with minor corrections, of the 1963 original, with a new preface and bibliography.
8. F.-J. DELVOS, *d-variate Boolean interpolation*, J. Approx. Theory, 34 (1982), pp. 99–114.
9. J. DICK, F. Y. KUO, AND I. H. SLOAN, *High-dimensional integration: the quasi-Monte Carlo way*, in Acta numerica, 2013, vol. 22 of Acta Numer., Cambridge Univ. Press, Cambridge, 2013, pp. 133–288.
10. G. E. FASSHAUER, *Solving differential equations with radial basis functions: multilevel methods and smoothing*, Adv. Comput. Math., 11 (1999), pp. 139–159. Radial basis functions and their applications.
11. G. E. FASSHAUER AND J. W. JEROME, *Multistep approximation algorithms: improved convergence rates through postconditioning with smoothing kernels*, Adv. Comput. Math., 10 (1999), pp. 1–27.

12. M. S. FLOATER AND A. ISKE, *Multistep scattered data interpolation using compactly supported radial basis functions*, J. Comput. Appl. Math., 73 (1996), pp. 65–78.
13. J. GARCKE AND M. GRIEBEL, *On the parallelization of the sparse grid approach for data mining*, in Large-Scale Scientific Computations, Third International Conference, LSSC 2001, Sozopol, Bulgaria, S. Margenov, J. Wasniewski, and P. Yalamov, eds., vol. 2179 of Lecture Notes in Computer Science, Springer, 2001, pp. 22–32. also as SFB 256 Preprint 721, Universität Bonn, 2001.
14. E. GEORGIOULIS, J. LEVESLEY, AND F. SUBHAN, SIAM J. Sci. Comput., 35 (2013), pp. 815–831.
15. M. GRIEBEL, M. SCHNEIDER, AND C. ZENGER, *A combination technique for the solution of sparse grid problems*, in Iterative methods in linear algebra (Brussels, 1991), North-Holland, Amsterdam, 1992, pp. 263–281.
16. S. J. HALES AND J. LEVESLEY, *Error estimates for multilevel approximation using polyharmonic splines*, Numer. Algorithms, 30 (2002), pp. 1–10.
17. A. ISKE, *Hierarchical scattered data filtering for multilevel interpolation schemes*, in Mathematical methods for curves and surfaces (Oslo, 2000), Innov. Appl. Math., Vanderbilt Univ. Press, Nashville, TN, 2001, pp. 211–221.
18. A. ISKE AND J. LEVESLEY, *Multilevel scattered data approximation by adaptive domain decomposition*, Numer. Algorithms, 39 (2005), pp. 187–198.
19. F. Y. KUO, C. SCHWAB, AND I. H. SLOAN, *Quasi-Monte Carlo methods for high-dimensional integration: the standard (weighted Hilbert space) setting and beyond*, ANZIAM J., 53 (2011), pp. 1–37.
20. G. N. MILSTEIN AND M. V. TRETYAKOV, *Stochastic numerics for mathematical physics*, Scientific Computation, Springer-Verlag, Berlin, 2004.
21. F. J. NARCOWICH, R. SCHABACK, AND J. D. WARD, *Multilevel interpolation and approximation*, Appl. Comput. Harmon. Anal., 7 (1999), pp. 243–261.
22. F. NOBILE, R. TEMPONE, AND C. G. WEBSTER, *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal., 46 (2008), pp. 2309–2345.
23. A. SCHREIBER, *The method of Smolyak in multivariate interpolation*, PhD thesis, der Mathematisch-Naturwissenschaftlichen Fakultäten, der Georg-August-Universität zu Göttingen, 2000.
24. S. A. SMOLYAK, *Quadrature and interpolation of formulas for tensor product of certain classes of functions*, Soviet Math. Dokl., 4 (1963), pp. 240–243.
25. S. L. SOBOLEV AND V. L. VASKEVICH, *The theory of cubature formulas*, vol. 415 of Mathematics and its Applications, Kluwer Academic Publishers Group, Dordrecht, 1997. Translated from the 1996 Russian original and with a foreword by S. S. Kutateladze, Revised by Vaskevich.
26. V. N. TEMLYAKOV, *Approximation of functions with bounded mixed derivative*, in Proc. Steklov Institute Math, 1989., AMS, 1989.
27. H. WENDLAND, *Numerical solution of variational problems by radial basis functions*, in Approximation theory IX, Vol. 2 (Nashville, TN, 1998), Innov. Appl. Math., Vanderbilt Univ. Press, Nashville, TN, 1998, pp. 361–368.